

(2)

2

(1)

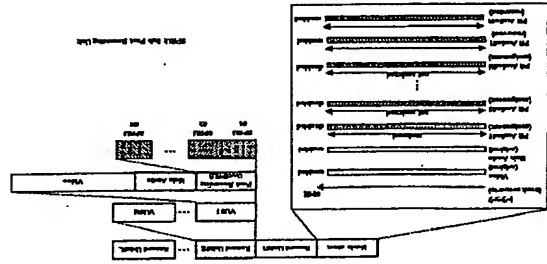
(19) 日本国特許庁 (J P) (12) 公開特許公報 (A) (11) 特許出願公開番号
特開2002-373480
(P2002-373480A)
(43) 公開日 平成14年12月26日 (2002.12.26)

(51) Int. Cl. G 11 B 27/036 20/10	識別記号 F I G 11 B 20/10	発明者 G 5 C 0 5 3 3 1 1 5 D 0 4 4 3 2 1 Z 5 D 1 1 0	発明者 チー・ワ・ド (参考) G 5 C 0 5 3 3 1 1 5 D 0 4 4 3 2 1 Z 5 D 1 1 0
20/12	20/12	3 1 1 3 2 1	3 1 1 3 2 1 Z 5 D 1 1 0

(21) 出願番号 特開2001-179418 (P2001-179416)	(71) 出願人 000005049 シャープ株式会社 大阪府大阪市阿倍野区長池町2番22号
(22) 出願日 平成13年6月14日 (2001.6.14)	(72) 発明者 木山 次郎 大阪府大阪市阿倍野区長池町2番22号 シャープ株式会社内 岩野 裕利 大阪府大阪市阿倍野区長池町2番22号 シャープ株式会社内 (74) 代理人 100102277 弁理士 佐々木 晴康 (外2名)

(54) 【発明の名称】 データ記録方法及びデータ記録装置ならびに記録媒体

(57) 【要約】
【課題】 複数のユーザプログラムからのアフレコを可能にする。
【解決手段】 映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータを格納するための第1の領域で構成される第1のユニットと、1個以上の前記第1のユニットを管理する第1のプログラムとを記録媒体に記録するデータ記録方法であって、前記第1のプログラムが、前記第1の領域の割り当て状態の管理情報を有するものである。



【特許請求の範囲】
【請求項1】 映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータを格納するための第1の領域で構成される第1のユニットと、1個以上の前記第1のユニットを管理する第1のプログラムとを記録媒体に記録するデータ記録方法であって、
前記第1のプログラムは、前記第1の領域の割り当て状態の管理情報を有することを特徴とするデータ記録方法。
【請求項2】 前記第1のプログラムは、前記第1の領域からその部分区間である第2のユニットを第2のプログラムに割り当てることを特徴とする前記請求項1に記載のデータ記録方法。
【請求項3】 前記第1のプログラムと前記第2のプログラムとが同一プログラムであることを特徴とする前記請求項2に記載のデータ記録方法。
【請求項4】 前記第2のプログラムは、割り当てられた前記第2のユニットの使用状態管理情報を有することを特徴とする前記請求項2乃至3のいずれかに記載のデータ記録方法。
【請求項5】 前記第2のプログラムは、前記第2のプログラムに割り当てられた前記第2のユニットの領域管理情報を有することを特徴とする前記請求項2乃至4のいずれかに記載のデータ記録方法。
【請求項6】 前記第2のユニットの割り当て状態の管理情報を更新し、前記第2のプログラムの作成時または修正時または削除時に行うことを特徴とする前記請求項2乃至5のいずれかに記載のデータ記録方法。
【請求項7】 前記第2のユニットの割り当て状態の管理情報を更新し、前記第2のプログラムへの前記第2のデータの記録時に行うことを特徴とする前記請求項2乃至6のいずれかに記載のデータ記録方法。
【請求項8】 前記第2のユニットが、固定長である第3のユニットで構成されることを特徴とする前記請求項2乃至7のいずれかに記載のデータ記録方法。
【請求項9】 前記第1のユニットが、独立再生可能な第4のユニットで構成されることを特徴とする前記請求項1乃至8のいずれかに記載のデータ記録方法。
【請求項10】 前記第4のユニットが、前記第1のデータのみで構成されることを特徴とする前記請求項9に記載のデータ記録方法。
【請求項11】 映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータを格納するための第1の領域で構成される第1のユニットと、1個以上の前記第1のユニットを管理する第1のプログラムとを記録媒体に記録するデータ記録装置であって、
前記第1のプログラムは、前記第1の領域の割り当て状態の管理情報を有することを特徴とするデータ記録装置。

【請求項12】 映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータを格納するための第1の領域で構成される第1のユニットと、1個以上の前記第1のユニットを管理する第1のプログラムとを記録される記録媒体であって、
前記第1のプログラムは、前記第1の領域の割り当て状態の管理情報を有することを特徴とする記録媒体。
【発明の詳細な説明】
【0001】
【発明の属する技術分野】 本発明は、映像データ、音声データをハードディスク、光ディスク等のランダムアクセス可能な記録媒体に対して記録・再生するデータ記録方法及びデータ記録装置に関するものである。
【0002】
【従来の技術】 ディスクメディアを用いたビデオや音声のデジタル記録再生装置が普及しつつある。それらにおいて、テープメディアと同様アプラーコーディング（アフレコ）機能を安価に実現する技術が求められていた。アフレコ機能は、既に記録したオーディオやビデオに対して、後から情報、特にオーディオを追記する機能である。
【0003】 ディスクメディアでアフレコ機能を実現している従来技術として、特開平5-234084号公報がある。この技術は、プログラム再生期間よりデータの読み込みが短いことを利用して、現在再生しているディスクからメモリにデータを読み込んでから次のデータを読み込むまでの間に、入力されたアフレコ音声データをディスクに書き込むというもので、ディスク記録再生手段が1つであっても、アフレコを実現することが可能である。
【0004】 ここでプログラム再生期間とは、ビデオや音楽などプログラムそれぞれが持つ固有の再生期間のことである。例えば1分間のビデオは、再生手段が変わったとしても1分間で再生されなければならない正確に再生されたとは言えない。
【0005】 従来技術におけるディスクの記録フォーマットを図38に示す。ディスクはEC（エラー・コレクション・コーディング）ブロックの列で構成される。ECブロックは符号化を行う際の最小単位であり、データに加えエラー補正用のパリティが付加され、符号化が行われている。データを読み込む際は、この単位で読み込み誤り訂正をしてから、必要なデータを取り出す。
【0006】 一方、データを書き換える際は、まずECブロック単位で読み込み、誤り訂正をしたデータに対して、必要な部分を書き換え、再度誤り符号の付与を行い、ディスクに記録を行なう。このことは、ハイパスを換える場合でも、そのバイトが含まれるECブロック全体を読み込み書き込む必要があることを意味する。
【0007】 ビデオやオーディオはECブロック中で、図38(b)に示すように、アフレコオーディオブロッ

(3)

4

刻14で終了すると、時刻14から領域s15～s17のデータをディスクから読み込む。このように、以下同様の処理を繰り返す。

【0016】 上述の従来技術では、情報圧縮を行うことにより、データの再生時間よりも読み込み時間が短くなることを利用し、記録再生手段を、記録と再生で時分割して利用することで、1つの記録再生手段だけでアフレコを実現している。尚、特開2001-118362号公報にも、同様の技術が開示されている。

10

【0017】

【発明が解決しようとする課題】 ここで、テープメディアにはないディスクメディアにおける特徴機能として、非破壊編集機能あるいはノンリニア編集機能と呼ばれる機能がある。この機能は、ディスク上に記録したAVストリームを移動あるいはコピーすることなく、AVストリームの任意の区間を任意の順番で再生できる、というものの、AVストリームのどこからどこまでという順番で再生するかを示す情報(再生管理情報)を作り、その情報に従って再生することを実現される。

20

【0018】 非破壊編集機能は単に再生順番を変えただけではなく、同一のシーンを複数のユーザープログラムで作る、つまり同一のシーンを複数のユーザープログラムで共有することが可能となる。このときのユーザーの要望としては、それぞれのユーザープログラムで個別にアフレコがしたい、というものがある。

【0019】 例えば、ユーザープログラムAではBGMをアフレコし、同じシーンを共有するユーザープログラムBではアナウンスをアフレコしたい、といった要望である。

【0020】 しかしながら、従来技術においては、非破壊編集によりあるシーンを複数のユーザープログラムで共有した場合、あるユーザープログラムにおいてアフレコした音声があるユーザープログラムでも再生されてしまう、また、あるユーザープログラムAにおいてアフレコした音声に、別のユーザープログラムBで上書きしてしまう、ユーザープログラムAのアフレコ結果を消してしまう、という問題があった。

【0021】 本発明は、上記課題に鑑みてなされたものであり、複数のユーザープログラムからのアフレコを可能にすることを目的とする。

40

【0022】

【課題を解決するための手段】 本願の第1の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納するための第1の領域で構成される第1のユニットと、1個以上の前記第1の領域に記録する第1のプログラムとを記録媒体に記録する第1の記録方法であって、前記第1のプログラムが、前記第1の領域の割り当て状態の管理情報を有することを特徴とする。

50

【0023】 本願の第2の発明は、前記第1のプログラムが、前記第1の領域からその部分区間である第2のユ

(4)

5

ニットを第2のプログラムに割り当てておくことを特徴とする。

【0024】 本願の第3の発明は、前記第1のプログラムと前記第2のプログラムとが同一プログラムであることを特徴とする。

【0025】 本願の第4の発明は、前記第2のプログラムが、割り当てられた前記第2のユニットの使用状態管理情報を有することを特徴とする。

【0026】 本願の第5の発明は、前記第2のプログラムが、前記第2のプログラムに割り当てられた前記第2のユニットの領域管理情報を有することを特徴とする。

【0027】 本願の第6の発明は、前記第2のユニットの割り当て状態の管理情報の更新を、前記第2のプログラムの作成時または修正時または削除時に行うことを特徴とする。

【0028】 本願の第7の発明は、前記第2のユニットの割り当て状態の管理情報の更新を、前記第2のプログラムの前記第2のデータの記録時に行うことを特徴とする。

【0029】 本願の第8の発明は、前記第2のユニットが、固定域である第3のユニットで構成されることを特徴とする。

【0030】 本願の第9の発明は、前記第1のユニットが、独立再生可能な第4のユニットで構成されることを特徴とする。

【0031】 本願の第10の発明は、前記第4のユニットが、前記第1のデータのみで構成されることを特徴とする。

【0032】 本願の第11の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生されるデータを格納するための第1の領域で構成される第1のユニットと、1個以上の前記第1のユニットを管理する第1のプログラムとを記録媒体に記録する第1の記録装置であって、前記第1のプログラムが、前記第1の領域の割り当て状態の管理情報を有することを特徴とする。

【0033】 本願の第12の発明は、映像又は音声からなる第1のデータと、前記第1のデータと同期して再生される第2のデータとを格納するための第1の領域で構成される第1のユニットと、1個以上の前記第1のユニットを管理する第1のプログラムとを記録媒体に記録する第1の記録媒体であって、前記第1のプログラムが、前記第1の領域の割り当て状態の管理情報を有することを特徴とする。

【発明の実施形態】 以下、本発明の実施形態について、図面を参照しながら詳細に説明する。

【0035】 <システム構成> 図1は本発明形態において共通に用いる、アフレコ可能なビデオディスクレコーダの構成である。図1に示すように、この装置は、バス100、ホストCPU101、RAM102、ROM103、ユーザーインター

6

フェース104、システムクロック105、光ディスク106、ビデオアップ107、ECCデコード108、ECCエンコード109、再生用バッファ110、記録/アフレコ用バッファ111、出マルチプレクサ112、マルチプレクサ113、多重化用バッファ114、オーディオデコード115、ビデオデコード116、オーディオエンコード117、ビデオエンコード118および図示されないカメラ、マイク、スピーカ、ディスプレイ等で構成される。

【0036】 ホストCPU101は、バス100を通じてデマルチプレクサ112、マルチプレクサ113、ビデオアップ107、デコード115、オーディオデコード116、オーディオエンコード117、ビデオエンコード118との通信を行う。

【0037】 再生時には、光ディスク106からピックアップ107を通じて読み出されたデータは、ECCデコード108によって誤り訂正され、再生用バッファ110に一旦蓄えられる。デマルチプレクサ112はオーディオデコード115、ビデオデコード116からのデータ送信要求に従って、再生用バッファ中のデータをその種別によって適当なデータに振り分ける。

【0038】 一方、記録時には、オーディオエンコード117とビデオエンコード118によって圧縮符号化されたデータは多重化用バッファ114に一旦送られ、マルチプレクサ113によってAV多重化され記録/アフレコ用バッファ111に送られる。記録/アフレコ用バッファ111中のデータは、ECCエンコード109によって誤り訂正符号を付加されピックアップ107を通じて光ディスク106に記録される。

【0039】 オーディオデータの符号化方式にはMP3-Layer-IIを、ビデオデータの符号化方式にはMPEG-2をそれぞれ用いる。

【0040】 光ディスク106は、外周から内周に向かって螺旋状に記録再生の行われる脱着可能な光ディスクとする。2048byteを1セクタとし、誤り訂正のため16セクタでECCブロックを構成する。ECCブロック中のデータを書き換える場合、そのデータが含まれるECCブロック全体を読み込み、誤り訂正を行い、対象のデータを書き換え、再び誤り訂正符号を付加し、ECCブロックを構成し記録媒体に記録する必要がある。また、記録効率を上げるため2048(ゾーン角速度一定)を採用しており、記録領域は回転数の異なる複数のゾーンで構成される。

【0041】 <ファイルシステム> 光ディスク106上の各種情報を管理するためにファイルシステムを用いる。ファイルシステムには、バーナクルコンピュータ(PC)との相互運用を考慮してUDF(Universal Disk Format)を使用する。ファイルシステム上では各種管理情報やAVストリームはファイルとして扱われる。

【0042】 エネルギーアは、2048byteの論理ブロック(セクタと一対一対応)で管理される。各ファイルは、整

数値のエクステント(連続した論理ブロック)で構成さ

3

ク、オリジナルオーディオブロック、オリジナルビデオブロックの順に配置される。それぞれのブロックにはほぼ同じ時間に対応するアフレコオーディオ、オリジナルオーディオ、オリジナルビデオが含まれている。

【0008】 尚、オリジナルオーディオブロックとオリジナルビデオブロックを合わせてオリジナルブロックと呼ぶことにする。オリジナルプログラム(アフレコオーディオを記録する前の映像)を記録する際は、アフレコオーディオブロックにダミーのデータを書き込んでおく。

【0009】 次に、従来技術におけるアフレコ時の動作について、図3.9とともに説明する。図中、上段のグラフは、ディスクからの読み込みと再生や記録といった各処理の時間的な関係を示しており、矢印内の記号は中段のグラフの縦軸に対応し、処理対象となっているデータのディスク上の位置を表す。中段は、ディスク中のヘッドの位置を、下段のグラフはバッファメモリにおけるプログラムデータの割合を模式的に示したものである。

【0010】 ここではプログラムが、ディスク中のs11～s18への逆順的な領域に配置され、領域s11～s13、s13～s15、s15～s17がそれぞれECCブロックに対応し、領域s11～s12、s13～s14、s15～s16、s17～s18がそれぞれアフレコオーディオブロックに対応しているものとする。

【0011】 時刻t:1の時点ですでにs13までの領域がバッファメモリ108に格納されており、領域s11～s13に記録されていたデータがデコードされ再生されるとともに、そのデータのアフレコ音声の入力、エンコードが行われる。

【0012】 時刻t:2において、領域s13～s15のデータをディスクから読み込み、バッファメモリ及びアフレコバッファへの格納を行う。アフレコバッファは読み込んだECCブロックをそのまま記憶し、図3.8(h)と同様の構成をとる。

【0013】 時刻t:3は、時刻t:2の時点で行われていた領域s11～s13に記録されていたデータのデコード、再生が行われる。この領域s13～s15のデータのデコード、再生は時刻t:5まで行われる。

【0014】 時刻t:4までに入力されたアフレコ音声は、少なくとも時刻t:3までにエンコードが終了する。時刻t:3において、時刻t:4までに入力されたアフレコ音声をディスク媒体に記録する。このとき、領域s11にアクセスする際に、ディスクの回転待ちの時間を要するが、ディスクの読み書きの時間に比べると、短時間であるので、ここでは考慮しない。

【0015】 アフレコ音声のディスクへの書き込みは、時刻t:3～t:4で行われる。このディスクへの書き込みが時

(6)

れ、エクステンデント単位で分散して記録してもよい。空き領域は、Space Bitmapを用いて除算ブロック単位で管理される。

【0043】＜ファイルフォーマット＞AVストリーム管理のためのフォーマットとして、QuickTimeファイルフォーマットを用いる。QuickTimeファイルフォーマットとは、Apple社が開発したマルチメディアデータ管理用フォーマットであり、PCの世界で広く用いられている。

【0044】QuickTimeファイルフォーマットは、ビデオデータやオーディオデータ等(これらを総称してメディアデータとも呼ぶ)と管理情報とで構成される。両者を合わせてここでは、QuickTimeムービー(略してムービー)と呼ぶ。両者は同じファイル中に存在しても、別々のファイルに存在してもよい。

【0045】同じファイル中に存在する場合は、図2(a)に示すような構成をとる。各種情報はatomという共通の構造に格納される。管理情報はMovie atomという構造に格納され、AVストリームはMovie data atomという構造に格納される。尚、Movie atom中の管理情報には、メディアデータ中の任意の時間に対応するAVデータのファイル中での相対位置を導くためのテーブルや、メディアデータの属性情報や、後述する外部参照情報等が含まれている。

【0046】一方、管理情報とメディアデータを別々のファイルに格納した場合は、図2(b)に示すような構成をとる。管理情報はMovie atomという構造に格納されるが、AVストリームはatomには格納される必要はない。このとき、Movie atomはAVストリームを格納したファイルを「外部参照」している。という。

【0047】外部参照は、図2(c)に示すように、複数のAVストリームフォーマットに対して行うことが可能であり、この仕組みにより、AVストリーム自体を物理的に移動することなく、見かけ上編集を行ったように見せる、いわゆる「ノンリニア編集」が可能になる。

【0048】それでは、図3乃至図12を用いて、QuickTimeの管理情報のフォーマットについて説明する。まず、共通の情報格納フォーマットであるatomについて説明する。atomの先頭には、そのatomのサイズであるatom size、そのatomの種類別情報であるTypeが必ず存在する。Typeは4文字で区別され、例えばMovie atomではmovie、Movie data atomでは'mdat'となっている。

【0049】各atomは別のatomを含むことができる。すなわち、atom間には階層構造がある。Movie atomの構成を図3に示す。Movie header atomは、そのMovie atomが管理するムービーの全体的な属性を管理する。Track atomは、そのムービーに含まれるビデオやオーディオ等のトラックに関する情報を格納する。User-defined data atomは、独自に定義可能なatomである。

【0050】Track atomの構成を図4に示す。Track header atomは、個々のチャンネルのデータが格納されている

8

ader atomはそのトラックの全体的な属性を管理する。Edit list atomを含む。Edit list atomはNumber of entriesで指定される個数分の、Track duration、Media time、Media rateの値の組(エントリ)を持つ。各エントリは、トラック上で連続的に再生される区間に対応し、そのエントリの再生時間順に並んでいる。

【0051】Track header atomの構成を図5に示す。ここでは、後での説明に必要なもののみ説明する。flagは属性を示すフラグの集合であり、代表的なものとしては、Track enabledフラグがあり、このフラグが1であれば、そのトラックは再生され、0であれば再生されない。layerはそのトラックの空間的な優先度を表しており、画像を表示するトラックが複数あれば、layerの値が小さいトラックほど画像が前面に表示される。

【0052】Media atomの構成を図6に示す。Media header atomは、そのMedia atomの管理するメディアデータに関する全体的な属性等を管理する。Handler reference atomは、メディアデータをどのデコーダでデコードするかを示す情報を格納する。Media information atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。

【0053】Media information atomの構成を図7に示す。Media information header atomは、ビデオやオーディオ等メディア固有の属性情報を管理する。Handler reference atomは、Media atomの項で説明した通りである。Data information atomは、そのQuickTimeムービーが参照するメディアデータを含むファイルの名前を管理するatomである(data reference atom)をサンプルと呼ぶ。個々のトラック毎に、サンプルには再生時間順に1から番号(サンプル番号)がついている。

【0054】次にSample table atomについて説明するが、その前に、QuickTimeにおけるデータの管理方法について、図8を用いて説明する。QuickTimeでは、データの最小単位(例えばビデオフレーム)をサンプルと呼ぶ。個々のトラック毎に、サンプルには再生時間順に1から番号(サンプル番号)がついている。

【0055】また、QuickTimeフォーマットでは、個々のチャンネルの再生時間表およびデータサイズを管理している。また、同一トラックに属するサンプルが再生時間順にファイル中で連続的に配置された領域をチャンネルと呼ぶ。チャンネルにも再生時間順に、1から番号がついている。

【0056】さらに、QuickTimeフォーマットでは、個々のチャンネルのファイル先頭からのアドレスおよび個々のチャンネルが含むサンプル数を管理している。これらの情報に基づき、任意の時間に対応するサンプルの位置を求めることができる。

【0057】Sample table atomの構成を図9に示す。Sample description atomは、個々のチャンネルのデータフォーマット(Data format)やサンプルが格納されている

9

ファイルのチャンネルのインデックス等を管理する。Time-to-sample atomは、個々のサンプルの再生時間を管理する。

【0058】Sync sample atomは、個々のサンプルのうち、デコード開始可能なサンプルを管理する。Sample-to-chunk atomは、個々のチャンネルに含まれるサンプル数を管理する。Sample size atomは、個々のサンプルのサイズを管理する。Chunk offset atomは、個々のチャンネルのファイル先頭からのアドレスを管理する。

【0059】Edit atomは、図10に示すように、1個のEdit list atomを含む。Edit list atomはNumber of entriesで指定される個数分の、Track duration、Media time、Media rateの値の組(エントリ)を持つ。各エントリは、トラック上で連続的に再生される区間に対応し、そのエントリの再生時間順に並んでいる。

【0060】Track durationはそのエントリが管理する区間のトラック上での再生時間、Media timeはそのエントリが管理する区間の先頭に対応するメディアデータ上の位置、Media rateはそのエントリが管理する区間の再生スピードを表す。尚、Media timeが1の場合は、そのエントリのTrack duration分、そのトラックでのサンプルの再生を停止する。この区間のことをempty editと呼ぶ。

【0061】図11にEdit listの使用例を示す。ここでは、Edit list atomの内容が図11(a)に示す内容であり、さらにサンプルの構成が図11(b)であったとする。尚、ここでは番目のエントリのTrack durationを(i)、Media timeを(t)、Media rateをR(t)とする。このとき、実際のサンプルの再生は図11(c)に示す順に行われる。このことについて簡単に説明する。

【0062】まず、エントリ#1はTrack durationが13000、Media timeが20000、Media rateが1であるため、そのトラックの先頭から13000の区間はサンプル中の時刻20000から33000の区間を再生する。次に、エントリ#2はTrack durationが5000、Media timeが1であるため、トラック中の時刻13000から18000の区間、何も再生を行わない。

【0063】最後に、エントリ#3はTrack durationが1000、Media timeが0、Media rateが1であるため、トラック中の時刻18000から28000の区間において、サンプル中の時刻0から10000の区間を再生する。

【0064】図12にUser-defined data atomの構成を示す。このatomには、QuickTimeフォーマットで定義されていない独自の情報を任意に格納することができる。1個の独自の独自情報は1個のエントリで管理され、1個のエントリはSizeとTypeとUser dataで構成される。Sizeはそのエントリ自体のサイズを表し、Typeは独自情報をそれぞれ区別するための識別情報、User dataは実際のデータを表す。

【0065】＜インデックス・ファイル＞ディレクトリに

10

含まれるQuickTimeムービーを管理するため、AVインデックス・ファイルという特別のQuickTimeムービー・ファイルをディレクトリ内に1個置く。

【0066】AVインデックス・ファイルには、ディレクトリ内のファイル(QuickTimeムービーやQuickTimeムービーから参照されている静止画像)に関するサムネイルや各種属性が登録されている。各種属性の中には、そのファイルが外部参照されている回数を示すlink countがある。

【0067】link countを参照することで、そのファイル参照しているファイルがあるかどうかを容易に知る事ができ、他から参照されているファイルを不用意に削除してしまふことを防ぐことができる。

【0068】＜実施例1＞本発明の第1の実施例について、図13乃至図29を用いて説明する。

【0069】＜AVストリームの形態＞まず、本実施例におけるAVストリームの構成について、図13乃至図15を用いて説明する。AVストリームは複数のRecord Unit(RU)で構成される。RUはディレクトリ上で連続的に格納する単位である。RUの長さは、AVストリームを構成するRUをどのようにディレクトリ上に配置してもシーマレス再生(再生中に途や音が途切れないで再生できること)やリアルタイムアフレコ(アフレコ対象のビデオをシーマレス再生しながらオーディオを格納することが保証されるように設定される。この設定方法については後述する。【0070】また、RU境界がEOCブロック境界と一致するようにストリームを構成する。RUのこれらの性質によって、AVストリームをディレクトリに格納した後も、シーマレス再生を保証したまま、ディレクトリ上でRU単位の配置を容易に変更できる。

【0071】RUは、整数個のVideo Unit(VU)で構成する。VUは単位再生可能な単位であり、そのことから再生の際のエントリ・ポイントとなりうる。VUの構成は、アフレコに対応したストリーム(アフレコ対応ストリーム)とアフレコには対応しないストリーム(アフレコ非対応ストリーム)とで異なる。

【0072】まず、アフレコ非対応ストリームにおけるVU構成を図14に示す。VUは、1秒程度のビデオデータ(例えば15フレーム)を格納した整数個のGOP(グループ・オブ・ピクチャ)とそれと同じ時間に再生されるメインオーディオデータとを格納した整数個のAMU(オーディオ・アクセス・ユニット)で構成される。

【0073】尚、GOPは、MPEG-2ビデオ規格における圧縮の単位であり、複数のビデオフレーム(典型的には15フレーム程度)で構成される。AMUはMPEG-1 Layer II規格における圧縮の単位で、1152点の音波形サンプル点により構成される。サンプリング周波数が48kHzの場合、AMUあたりの再生時間は0.024秒となる。VU中ではAMU期間再生のために必要となる遅延を小さくするためAMU、GOPの順に配置する。

(7)

11

【0074】また、VU単位で独立再生可能なように、VU中のビデオデータの先頭にはSequence Header (SH)を、末尾にはSequence End Code (SEC)を置く。VUの再生時間は、VUに含まれるビデオフレーム数にビデオフレーム周波数をかけたものと定数とする。また、VUを整数個組み合わせてRUを構成する場合、RUの始端をECCブロック境界に合わせるため、VUの末尾を0で埋める。

【0075】一方、アフレコ対応ストリームにおけるVUの構成は、図15に示すとおりである。アフレコ非対応ストリームにおけるVUの先頭にビデオおよびメインオーディオデータと同期して再生を行うアフレコ (サブオーディオ)データ格納するための領域としてPost Recording Unit (PRU)を設け、その中をN個の固定サイズのSubPost Recording Unit (SPRU)に分割している。

【0076】Nは、そのAVストリームに対して行いたいアフレコのバリエーションの最大数に設定する。尚、ここではPRUをメインオーディオを格納するための領域の前に置いているが、その逆でも構わない。

【0077】SPRUのサイズは、低いオーディオビットレート (例えば64kbit/s)に基づき確保する。このことによつて、アフレコを普発行わないが、極大に必要性が発生する、という多くのユーザにとって、ディスクの記録容量をわずかながらでも節約することは可能となる。

【0078】アフレコ入力力人間の音声であることが多いため、低いビットレートであっても、たいいの場合には音質的に問題ない、SPRUのサイズを固定にすることによって、後述する領域の割り当てや解放を繰り返した場合であっても、未割り当ての小さな領域が分散するいわゆるフラグメンテーションが起きづらい、という利点がある。

【0079】尚、ここではアフレコの対象はビデオとメインオーディオ、アフレコにおいて記録するデータはサブルービメントに限定されるものではない。

【0080】<AVストリーム管理方法>AVストリームの管理方法は、前述のQuickTimeファイルフォーマットをベースにしている。図16にQuickTimeによるAVストリーム管理形態を示す。図16(a)はアフレコ非対応の場合を示している。AMU、Sequence headerからSequence endcodeまでのビデオデータをそれぞれサンプルで管理し、VU中のメインオーディオとビデオの塊をそれぞれ1チャンクに対応させる。

【0081】図16(b)はアフレコ対応ストリームの構成を示している。基本的にはアフレコ非対応ストリームの場合と同じであるが、ISPRUを1チャンクで管理している点で異なる。これは、特定のSPRUのみを抜き出すことを容易にするためである。

【0082】図17にアフレコ対応ストリームで構成されるオリジナルプログラムデータの構成を示す。また、図18にアフレコ対応ストリームで構成されるユーザプログラ

12

Mの構成を示す。尚、ここでオリジナルプログラムとMは、ディスクに記録されたAVストリームを管理するQuickTime管理情報を指す。一方、ユーザプログラムとは、オリジナルプログラムによって管理されるAVストリームを参照するQuickTime管理情報を指す。

【0083】前述のように、管理情報であるMovie atomとAVストリームから構成される、Movie atomにはビデオデータを管理するためのビデオトラック、メインオーディオデータを管理するためのメインオーディオトラック、アフレコオーディオデータを管理するためのアフレコオーディオトラックが含まれる。

【0084】アフレコオーディオトラックには、オリジナルプログラムが各プログラムへの割り当て状況を管理するためのトラック (割り当て管理トラック、図17中のAssignment)と、自プログラムにアフレコ用として確保されている領域を管理するためのトラック (領域管理トラック、図17及び図18中のReserved)と、実際にアフレコ用として利用されている領域を管理するためのトラック (使用管理トラック、図17及び図18中のRecorded)がある。

【0085】まず、割り当て管理トラックについて説明する。割り当て管理トラックは、AVストリーム記録時に確保したPRU中のSPRUに対応する数(ここではN)だけオリジナルプログラムに作成される。各割り当て管理トラックは、Sample table atomによって、そのトラックが管理する各SPRUをチャンクおよびサンプルに対応させ、アドレスやサイズや対応する再生時間を管理する。

【0086】さらに、各SPRUが既割り当て済みか未割り当てかを、Edit list atomで管理する。具体的に、既割り当ての区間は再生対象とし、未割り当ての区間はEmpty listによって再生対象から外すことで、各SPRUの割り当て状態を管理する。

【0087】このトラックを参照することで、異なるプログラム間でのアフレコデータの上書きを防止することができ、仮にこのトラックが無かった場合、アフレコデータの上書きを防止するためには、同じPRUを参照している可能性のあるユーザプログラムの管理情報をすべてチェックし、未使用の領域を捨廃する必要があることになり、シーク時間の長い光ディスクにおいては処理時間の面で問題となる。

【0088】次に、領域管理トラックについて説明する。領域管理トラックは、そのプログラムに割り当てられたアフレコ領域を管理するためのトラックである。このトラックがあるため、割り当て元のオリジナルプログラムの参照することなく、アフレコ時にデータをどこに記録すれば良いかが分かる。

【0089】領域管理トラックでは、基本的には割り当てられた領域のアドレスとサイズと対応する再生時間を管理すればよいが、ここではデータを格納すべき位置をアフレコ時に計算する必要があるように、領域管理トラ

(8)

13

ックのSample table atomには、既にデータが記録されていると想定して値を格納しておく。

【0090】最後に、使用管理トラックについて説明する。使用管理トラックは、領域管理トラックによって割り当てられた領域のうち、実際にアフレコによって格納されたサンプルをSample table atomで管理し、そのうち実際の再生対象となっている区間をEdit list atomによって指定する。

【0091】アフレコされた区間をSample tableではなく、Edit listで指定する方法も考えられるが、アフレコデータに関してもEdit listで再生区間の指定を可能にしたため、アフレコされた区間をSample tableで管理している。使用管理トラックは、再生時間がオーバーラップしない限り1個のプログラムに複数存在しても構わない。このトラックにより、意図しないアフレコデータの書き込みを防ぐことが可能になる。

【0092】このような管理方法においては、同一のアフレコ領域に対応するトラックが領域管理トラックと使用管理トラックの最大3トラックが存在するため、同一データが最大三箇に再生されるおそれがある。それを避けるため、領域管理トラックおよび割り当て管理トラックはTrack header atomのTrack enabledフラグを0にセットし、使用管理トラックのTrack enabledフラグの方のみを1にセットしている。

【0093】また、上記のトラック属性を管理するため、図19に示すように、QuickTimeに属性を追加している。追加には前述のTrack atom中のUser-defined atomを用いており、独自のAtom typeであるtrack property ('tkpt')を定義してある。このときtrack propertyの値が'orig'ならオリジナル・トラック、'pasn'なら割り当て管理トラック、'prev'なら領域管理トラック、'pre'なら使用管理トラックを意味する。

【0094】尚、本実施例では、オリジナルプログラムにおいて割り当て管理トラックと領域管理トラックとの両方を用いているが、1番目のSPRU (図16(b)におけるSPRU1)をオリジナルプログラム用に必ず割り当てようになれば、領域管理トラックは不要となる。

【0095】<ディスク配置決定方法>まず、アフレコ対応ストリームにおける再生時間の決定方法について説明する。この決定方法では、機器間の互換性確保のため、基準となるデバイス (リファレンス・デバイス・モデル)と基準となるアプロキシム (リファレンス・モデル)とを定義し、次にそれらを用いてアフレコを行った際にシーマレム再生が破綻しないように再生時間を決める。

【0096】それではまず、リファレンス・デバイス・モデルについて、図20を用いて説明する。リファレンス・デバイス・モデルは1個のピクチャップとそれにつながるECCエンコーダ・デコーダ501、トラックバッファ502、デマルチプレクサ503、アフレコ用バッファ504、

14

オーディオエンコーダ509、ビデオバッファ505、オーディオバッファ506、ビデオデコーダ507、オーディオデコーダ508で構成される。

【0097】本モデルでは、ピクチャップが1個であるため再生用データのディスクからの読み出しとアフレコ用データのディスクへの記録は時分割で行う。ディスクから再生用データを読み出す際、PRUも含めて読み出す。読み出されたPRUを含むECCブロック (PRUブロック)は、トラックバッファ502からアフレコ用バッファ504に送られる。

【0098】オーディオエンコーダ509はAMU周波で、アフレコ用バッファ504に出力する。この出力によって、アフレコ用バッファ504中の対応するPRUブロックを上書きする。アフレコデータの記録は、PRUブロックを所定のECCブロックに記録することで行う。

【0099】本モデルにおいて、PRUブロックをトラックバッファ502からアフレコ用バッファ504に送ることを想定しているのは次の理由による。本実施形態におけるAVストリームではPRU境界とECCブロック境界は一致しないため、PRU境界を含むECCブロックにはPRUのデータだけではない、その他のデータ (直前のVUのビデオデータや同一のVUのオーディオデータ)が含まれる。

【0100】したがって、PRUにデータを記録する際には、PRU境界を含むECCブロックをメモリに一旦読み出す必要がある。PRUを記録する直前にメモリに読み出すということも考えられるが、再生用データ読み出し時にPRU境界を含むECCブロックを必ず読み出していることから、再生用データ読み出し時に読み出したPRUを含むECCブロックを一時的にアフレコ用バッファ504に保持しておくことで、PRU境界を含むECCブロックの再度読み出しを省略している。

【0101】本モデルにおけるシーマレム再生は、VUのデコード開始時にトラックバッファ502上に少なくとも1個VUが存在すれば保証されるものとする。オーディオアフレコデータのECCエンコーダ501へのデータの入力速度およびECCデコーダ501からデータの出力速度はfsとす

る。

【0102】また、アクセスによる読み出し、記録の停止する最大時間をTaとする。さらに、短いアクセス (100トラック程度)に要する時間をTtとする。尚、これら期間にはシーク時間、回転待ち時間、アクセス後に最初にディスクから読み出したデータがECCから出力されるまでの時間が含まれる。本実施例ではfs=20Mbps、Ta=1秒、Tt=0.2秒とする。

【0103】次に、リファレンス・アフレコ・アルゴリズムについて、図21を用いて説明する。尚、図21中の(1)から(9)までの番号は、以下の説明中の(1)から(9)までの番号に対応する。

【0104】アルゴリズムの概要は次の通りである。

(1) 再生用データの読み出しを行う。(2) N番目のRUで

(11)

19

タが蓄積されているかどうかをチェックし(ステップ704)、蓄積されている間ステップ705からステップ708を繰り返す。蓄積されていれば、次に記録するディスク上の ECCプロットの空き状況をRAM上のSpace Bitmapを参照して調べる(ステップ705)

空きがなければ、17個のVUを記録可能な連続的な空き領域を探し(ステップ707)、その空き領域の先頭へビットマップを移動する(ステップ708)。次に、記録用バッファ111中のECCプロット分のデータをディスクに記録する(ステップ709)。

【0140】記録用バッファ111にデータが蓄積されていなければならないが、記録終了が指示されているかどうかをチェックし(ステップ709)、記録終了でなければステップ704を実行する。

【0141】記録終了が指示されていた場合、以下のステップを実行する。まず、記録用バッファ中の32KBに満たないデータに関して、末尾にダミーデータを付加し32KBにする(ステップ710)。次に、そのデータをディスク上に記録する(ステップ711～ステップ714)。最後に、RAM102上のQuickTime管理情報とファイルシステム管理情報を光ディスク108に記録する(ステップ715～ステップ716)。

【0142】以上の処理と並行するオーディオエンコーダ117、ビデオエンコーダ118やマルチプレクサ113の動作について説明する。それぞれエンコーダはマルチプレクサ113にエンコード結果を送り、マルチプレクサはそれらを多重化用バッファ114に格納する。1VU分のデータ、つまりGOPとそれに同期して再生されるAUが多量化用バッファ114に蓄積されたら、マルチプレクサ113は記録用バッファ111に1VUのデータを送る。

【0143】その際、マルチプレクサ113はそのVU中のAUの個数に応じて、指定されたビットレートのAUを格納可能なSPRUを5個多量化する。さらに、ポストCPU101に1VU分のデータがエンコードできたことを通知し、ポストCPU101はVUを構成するGOPやAUの数およびサイズを基にRAM102上のQuickTime管理情報を更新する。

【0144】光ディスク108に記録された直後のQuickTime管理情報における各トラックの状態を図29に示す。割り当て管理トラックはN個なら5個作成される。1個目の割り当て管理トラック(図中のPR Audio#1(assign))のみにEdit listに1つより全区間再生対象とし、それ以外の割り当て管理トラック(図中のPR Audio#2～5(assign))に関しては、Edit listを用いて全区間再生対象から外す。なぜなら、1個目の領域(SPRU#1)はそのオリジナルプログラム自身に割り当てられ、それ以外の領域は未割り当てであるためである。

【0145】また、領域管理トラック(図中のPR Audio#1(reserved))を作成する。

【0146】<ユーザープログラム作成時の処理>次に、オリジナルプログラムを外参照したユーザープログラム

20

を作成する際の手順を説明する。ここでは、作成するユーザープログラムが、図24に示すようにオリジナルプログラムの一部を抜き出して再生というものである場合を考える。

【0147】また、ユーザーによって、ユーザープログラムから参照するオリジナルプログラム中の区間が決定されていると仮定する。さらに、オリジナルプログラムに関する管理情報がRAM102上に読み込まれ、新録再生のユーザープログラムに関する管理情報用の領域がRAM102上に確保されていると仮定する。

【0148】まず、RAM102上のオリジナルプログラムに関する管理情報を参照し5個ある割り当て管理トラックをすべてチェックし、指定された区間が新規に割り当て可能かどうかを調べる。仮に未割り当て領域があった場合、以下の手順を進める。まず、その領域をこのユーザープログラム用に割り当てする。割り当てでは、未割り当ての領域に对应する割り当て管理トラックのEdit list中の区間を再生対象とすることで行う。

【0149】次に、指定された区間に対応するビデオトラックおよびメインオーディオトラックのSample tableをユーザープログラム管理情報にコピーする。そして、割り当てられた領域を管理するための領域管理トラックを作成する。

【0150】具体的には、割り当てられた領域に実際にデータを記録したと想定して、Sample table atomを構築する。一方、未割り当て領域がなかった場合、指定された区間に対応するビデオトラックおよびメインオーディオトラックのSample tableのコピーのみ行う。

【0151】最後に、RAM102上のオリジナルプログラムおよびユーザープログラムの管理情報を光ディスク106に記録する。

【0152】ユーザープログラム作成前のオリジナルプログラムの管理情報構成を図24(a)、作成したユーザープログラムの管理情報構成を図24(b)に示す構成であった場合、ユーザープログラム作成後オリジナルプログラムの管理情報の構成を図25に示す。

【0153】尚、本実施例では、1個のユーザープログラムは各知から1個のSPRUを割り当てようになっているが、より高いビットレートでアフレコしたい場合、複数個SPRUを連続的に割り当ててもよい。

【0154】<アフレコ時の処理>ユーザーからアフレコが指示された場合は、図26とともに説明する。尚、以下の処理はユーザープログラムの場合もオリジナルプログラムの場合も共通である。すでにアフレコの対象となるAVストリームに関するQuickTime管理情報はRAM102に読み込まれているとする。

【0155】まず、そのQuickTimeムービーがアフレコ対応ストリームで構成されているかを調べ、そうでなければユーザーにアフレコできないことを通知する(ステップ801)。

(12)

21

【0156】アフレコ開始位置を含むディスク上のVUの先頭から再生用データの読み出しを行う(ステップ802)。このとき、十分な再生時間分のデータを読み出すまで、ステップ802を繰り返す(ステップ803)。ここで十分とは、再生用データ読み出しの中断期間が最大の場合でも再生が途切れないだけのデータ量を意味する。

【0157】また、アフレコ用領域(ここではSPRU)を読み出した際には、アフレコ用領域を含むECCブロック(PRプロット)をアフレコ用バッファ111に送る。このとき、アフレコ用バッファ111中のPRUを管理するために、アフレコ用バッファ111中の各アフレコ用領域の再生開始時間(AVストリームの先頭からの相対時間)とアフレコ用バッファ111中でのアドレスの紐をテーブルとしてRAM102に保持する。

【0158】次に、ビデオエンコーダ118とオーディオエンコーダ115および、オーディオエンコーダ117を起動する(ステップ804)。オーディオエンコーダ117はサンプリングされた音声波形をAMUにエンコードし、AMUの周回でマルチプレクサ113に送る。その際に、各AMUについてAVストリームの先頭からの相対時間を付加する。

【0159】マルチプレクサ113は、AMUに付加された時間に基づき、AMUをアフレコ用バッファ111中のPRUブロックに格納する。RU中の最後のPRUブロックにAMUを最後まで格納し終わったら、ポストCPU101にPRUのエンコード終了を通知する。

【0160】次に、ユーザーからアフレコ終了を指示されていないかチェックする(ステップ805)。指示されないければ、PRUのエンコード終了の通知があるまで、ステップ802と同様に再生用データの読み出しを行う(ステップ809)。

【0161】マルチプレクサ部からPRUエンコード終了が通知されたら(ステップ805)、RAM102上のテーブルに保持しているそのRUに含まれるPRUの再生開始時間から、QuickTime管理情報を用いてそのPRUを記録すべき光ディスク106上のアドレス、つまり元々そのPRUが記録されていたアドレスを求める。そのアドレスにビットマップ107を移動させ(ステップ807)、そのPRUブロックを光ディスク107に記録する(ステップ808)。

【0162】アフレコ終了を指示されていれば、現在エンコード中のPRUのエンコード完了を待つ(ステップ810)。そのPRUの記録アドレスを求めビットマップを移動し(ステップ811)、PRUを記録する(ステップ812)。最後にQuickTime管理情報をディスクに記録する(ステップ813)。

【0163】尚、本実施例では、再生用データの読み出しを中断しPRUの記録を開始する際に、再生用バッファ110の占有量をチェックを行っているが、アフレコ時のジョブの耐性を高めるために、チェックを行った方がよいことは言うまでもない。ただし、この場合、PRU記録タイミングが遅れる分、より多くの再生用バッファ

22

110およびアフレコ用バッファ111の容量が必要となる。

【0164】アフレコ直後のオリジナルプログラムのQuickTime管理情報における各トラックの状態を図27に示す。アフレコによって使用管理トラックが1個(図27中のPR Audio#1(recorded))追加される。ユーザープログラムの場合も図28に示すように、オリジナルプログラムのと同様、アフレコによって使用管理トラックが1個(図28中のPR Audio#1(recorded))追加される。

【0165】<ユーザープログラム削除時の処理>ユーザーから、ユーザープログラムの削除が指示された場合の処理について説明する。尚、すでに削除指示されたユーザープログラムの管理情報がRAM102上に読み込まれているものとする。

【0166】まず、ユーザープログラムの管理情報中の領域管理トラックを調べ、そのユーザープログラムがアフレコ領域を割り当ててもらっているオリジナルプログラムをリストアップする。

【0167】次に、リストアップした各オリジナルプログラムに関して以下の処理を行う。まず、オリジナルプログラムの管理情報をRAM102に読み込む。次に、そのオリジナルプログラムから領域を割り当ててもらっている全サンプルについて、オリジナルプログラム中の対応するサンプルを検索する。尚、検索はサンプル同士のアドレスの比較によって行う。

【0168】検索が終了したら、そのオリジナルプログラムに関して、一致したサンプルに対応する区間のアフレコ領域の割り当てを解放する。具体的には割り当て管理トラックのEdit list atomを書き換え、その区間を再生対象から外す。最後に、ユーザープログラムを格納しているファイルに光ディスク106から削除し、オリジナルプログラムの管理情報を光ディスク106に記録する。

【0169】<オリジナル削除時の処理>ユーザーから、オリジナルプログラムの削除が指示された場合の処理について説明する。尚、すでにRAM102上にAVインデックス・ファイルが読み込まれているものとする。

【0170】まず、RAM102上のAVインデックス・ファイルを参照し、削除指定されたオリジナルプログラムを検索する。検索の結果見つかったファイルのlink countを参照し、link countが0であればそのオリジナルプログラムの削除を行う。

【0171】<再生時の処理>ユーザーから再生が指示された場合の処理を、図29とともに説明する。すでに再生の対象となるAVストリームに関するQuickTime管理情報はRAM102に読み込まれているとする。

【0172】光ディスク107上の再生指示されたVUの先頭から再生用データの読み出しを行う(ステップ901)。このとき、十分な再生時間分のデータを読み出すまで、ステップ901を繰り返す(ステップ902)。ここで十分とは、再生用データ読み出しの中断期間が最大の場合でも再生

(13)

23

が送切れないだけのデータ量を意味する。具体的には、AVデータの読み出しに伴う秒断のジャンプ量(最大1秒)を行った場合を想定し、1秒分のデータ量とする。
[0173] 次に、ビデオコーデック116およびオーディオコーデック115を起動する(ステップ903)。そして、ユーザから再生終了を指示されていないかチェックする(ステップ904)。指示されていないければ、再生用AVデータの読み出しを行う(ステップ905)。再生終了を指示されれば、終了する。

[0174] <バリエーション> 本実施例では、各VUに1個のPRUを置くというAVストリーム構成を用いているが、後述する第2の実施例におけるAVストリーム構成を用いてもよい。さらに、アフレコ用の領域を備えるAVストリームであればどのような形態でも適用可能であることは言うまでもない。

[0175] また、本実施例では、アフレコの対象はビデオとメインオーディオ、アフレコにおいて記録するデータはサブオーディオデータとしているが、特にこれに限定されるものではない。例えばアフレコにおいて記録するデータが静止画面データや制御データであっても構わない。

[0176] さらに、本実施例では、領域が未割り当てに割り当てたかとの区別にEdit listatomを利用したが、区別が可能であれば別種の情報を使っても構わないことは言うまでもない。例えば、各SPRUが未割り当てに割り当てたかを示す独自の管理情報を、割り当て管理トラックのSample table atomに導入することも考えられる。また、後述する第3の実施例のように、チャンネルのData formatを利用してもよい。

[0177] そしてまた、本実施例では、割り当て管理トラックにおいて各SPRUを1チャンネルおよびサンプルとしてSample table atomを構成しているが、実際にオーディオデータを格納し、それらを管理しているとしてSample table atomを構成してもよい。こうすることで、ユーザプログラム作成時に、割り当て管理トラックのSample table atomのコピーで領域管理トラックのSample table atomの作成が可能になる。

[0178] さらに、本実施例では、録画時ユーザプログラム作成時に各プログラムに領域を割り当てているが、後述の第2の実施例で説明するように、アフレコ時に割り当ててもよい。その場合、各プログラムに割り当てられている領域は既にデータが記録されることになるため、領域管理トラックを備える必要はない。

[0179] <実施例2> 次に、本発明の第2の実施例について説明する。本実施例は、上述した第1の実施例と類似するため、相違点に絞って以下説明を行う。

[0180] <AVストリームの形態> 本実施例におけるストリームの構成を図30に示す。第1の実施例と異なる点は、複数のVU毎にアフレコ用の領域(PRU)をまとめて確保している点である。このように、まとめて確保す

(14)

24

ることによって、個々のVUにPRUを分散させるよりも大きな連続領域が確保可能であり、領域割り当ての柔軟性が増し、後述する静止画面等の格納が容易になる利点がある。

[0181] VUの構成については、図14とともに上述した第1の実施例と同様であるため、説明を省略する。PRUの領域サイズは、ここではVUの再生時間にオーディオの最大ビットレートをかけたものとする。

[0182] <AVストリーム管理方法> AVストリームの管理方法は、VUの管理については第1の実施例と同様である。PRUの管理については、以下説明する。

[0183] 第1の実施例と同様、第2の実施例でもPRUを管理するために割り当て管理トラックと領域管理トラックと使用管理トラックを用いる。領域管理トラックと使用管理トラックについては第1の実施例と同様であるため、割り当て管理トラックについてのみ説明する。
[0184] 割り当て管理トラックは、オリジナルプログラムのみが存在し、割り当て管理トラックと未割り当て管理トラックの2種類がある。割り当て管理トラックは、PRU中の割り当て済み領域を管理するトラックであり、PRU中の未割り当て領域の境を1サンプルで構成される1チャンネルに対応させる。

[0185] さらに、その割り当て領域とRUとの間の時間的な対応が取れるように、その割り当て領域に一致するサンプルのdurationを対応するRUの再生時間に一致させる。QuickTimeにおいては、1トラック中に時間的に重なるチャンネルを複数置くことはできないため、同一PRU中の異なる割り当て領域は別トラック、すなわち別の割り当て管理トラックで管理することになる。

[0186] また、AVストリーム中のすべてのPRUにおいて、同一数の割り当て領域が存在するとは限らないため、割り当て管理トラックに属してAVストリーム中のすべての領域に属して管理対象の領域が存在するとは限らない。したがって、管理対象の領域が存在しない区間に属してはその区間の再生時間分Edit list中にempty itemを設ける。

[0187] PRU中の未割り当て管理トラックについて、PRU中の未割り当て領域について、割り当て管理トラックと同様の管理を行う。このような構成を取ることで、本実施例では第1の実施例と異なり、割り当て領域サイズを動的に設定可能となっている。

[0188] 図31に割り当て管理トラックの例を示す。ここではPRU中に割り当て領域3101、未割り当て領域3102、割り当て領域3103、未割り当て領域3104があったとする。

[0189] この場合、上述のごとく、割り当て領域3101は割り当て管理トラック3111、割り当て領域3103は割り当て管理トラック3113、未割り当て領域3102は未割り当て管理トラック3112、未割り当て領域3104は未割り当て管理トラック3114の区間T1~T2がそれぞれ管

(15)

25

理する。未割り当て管理トラック3115の区間T1~T2がempty editとなっているのは、そのトラックのその区間には管理対象の領域が存在しないためである。

[0190] また、上記のトラック属性を管理するため、第1の実施例と同様、Track atom中のUser-defined atomに独自のAtom typeである「tktp=track property」を定義し、未割り当て管理トラックを示す「p yes」と割り当て管理トラックを示す「pasn」を定義する。

[0191] <ディスク配列決定方法> アフレコ対応ストリームにおける再生時間の決定方法は、第1の実施例と同様に、機器間の互換性確保のため、基準となるデバイス(リファレンス・デバイス・モデルと基準となるアフレコアルゴリズム(リファレンス・アフレコ・アルゴリズム))を想定し、次にそれらを用いてアフレコを行った際にシームレス再生が破綻しないように再生時間間を定める。このうち、リファレンス・デバイス・モデルは第1の実施例と同様であるため説明を省略する。

[0192] 本実施例におけるリファレンス・アフレコ・アルゴリズムについて、図32を用いて説明する。図32中の(1)から(8)までの番号は、以下の説明中の(1)から(8)までの番号に対応する。アルゴリズムの概要は次の通りである。

[0193] (1) 再生用データの読み出しを行う。(2) N *
$$Te(i) \geq (Ta \times RstLy) / (Rs - Rv - Ra - 2Rp)$$
 が得られる。

[0198] つまり、アフレコ保証可能な再生時間下
$$T_{min} = (Ta \times RstLy) / (Rs - Rv - Ra - 2Rp)$$
 となる。

[0199] このとき、再生時間の上限値Tmaxを次のように設定する。上限値を設定するのは第1の実施例と同様の理由に基づく。

$$T_{max} = T_{min} \times T_{max} \dots \text{式 14}$$

<記録時の処理> ユーザから録画が指示された場合の処理を、図22とともに説明する。このとき記録するAVストリームは、ビデオのビットレートRv=50Mbps、メインオーディオのビットレートRa=256kbpsで、VU再生時間固定のアフレコ対応ストリームであるとする。

[0200] すでに、ファイルシステムの管理情報はRA [0201] Tmax=約0.6秒での条件を満たすのはTe(i)=4.5秒のときとなり、9個のVU毎にPRUが挿入されることになる。MPEG-1 audio layer-Iにおいて、ビットレート256kbpsのとき、AAUの再生時間Tafは0.024秒、サ

(16)

26

* 番目のPRUであるPRU#Nに対応するオーディオデータのエンコードが終了すると同時にPRU#Nへのアクセスを行う。(3) PRU#Nに対応するPRUブロックをディスクに記録する。(4) 元の読み出し位置に戻る。(5) 再生用データの読み出しを行う。読み出しの間にRU境界を越え、分割のジャンプが発生するが、そのまま読み出しを続ける。
[0194] (6) N+1番目のPRUであるPRU#N+1に対応するオーディオデータのエンコードが終了すると同時に、PRU#N+1へのアクセスを行う。(7) PRU#N+1に対応するPRUブロックをディスクに記録する。(8) 元の読み出し位置に戻る。以上の動作を繰り返す。

[0196] 本実施例においても、第1の実施例と同様、<式1>を満たすことで、シームレス再生を確保できる。このとき、Tr(i)は第1の実施例と同一である。一方、PRU全体のビットレートをRtとしたとき、
$$Tr(i) = T_{min} \times Rv / RstLy / Rs \dots \text{式 11}$$
 である。

[0196] ここで、PRUの全領域を記録すると想定してTr(i)を設定しているのは、最大の割り当てをしたとしてもリアルタイムアフレコを確保するためである。

[0197] このとき、<式1>に式3>と式11>を代入して、Te(i)で解くと、リアルタイムアフレコを確保可能なTe(i)の条件

$$\dots \text{式 12}$$

※ 上限値Tminは、

$$\dots \text{式 13}$$

サイズは768byteとなり、このときのPRUの領域サイズは、144384byteとなる。また、連続領域には9個のVUが含まれるようにする。

[0202] 9個のVUと1個のPRUを連続的に記録可能な空き領域を探す。具体的には
$$9 \times T_{max} \times (Ra + Rv) + 9 \times T_{av} \times Ra$$
、つまり24.88bit以上の連続的な空き領域をPRU102上のSpace Bitmapを参照して探す。存在しなければ録画を中止し、録画できないことをユーザに知らせる(ステップ702)。

[0203] オーディオエンコーダ117、ビデオエンコーダ118をそれぞれ起動する(ステップ703)。そして、録画用バッファにIECGブロック分(30KB)以上のデータが蓄積されているかどうかをチェックし(ステップ704)、蓄積されている間ステップ705からステップ708を繰り返す。

[0204] 蓄積されているれば、次に記録するディスク上のECGブロックの空き状況をRAM上のSpace Bitmapを参照して調べる(ステップ705)。空きがなければ、9個のVUとPRUを記録可能な連続的な空き領域を探し(ステップ710)、その空き領域の先頭へヘッドアップを移動する(ステップ708)。

(15)

27

【0205】記録用バッファ111中のI/ECCブロック分のデータをディスクに記録する(ステップ709)。記録用バッファ111にデータが蓄積されていなければ、記録終了が指示されているかどうかをチェックし(ステップ709)、記録終了でなければステップ704を実行する。

【0206】記録終了が指示されていた場合、以下のステップを実行する。まず、記録用バッファ中の32KBに満たないデータに随って、末尾にダミーデータを付加し32KBにする(ステップ710)。次に、そのデータをディスク上に記録する(ステップ711)。次に、RAM102上のQuickTime管理情報とファイルシステム管理情報をディस्क106に記録する(ステップ715)。ステップ716。

【0207】以上の処理と並行するオーディオエンコーダ117、ビデオエンコーダ118やマルチプレクサ113の動作について説明する。それぞれのエンコーダはマルチプレクサ113にエンコード結果を送り、マルチプレクサはそれらを多重化用バッファ114に格納する。

【0208】1VU分のデータ、つまり1GOPとそれと同期して再生されるAMUが多重化用バッファ114に蓄積されたら、マルチプレクサ113は記録用バッファ111に1VUのデータを送る。このとき、そのVUが9×14番目(1は0以上の整数)のVUであったと、上述のサイズを持ったPRUを先に記録用バッファ111に送る。

【0209】さらに、ホストCPU101に1VU分のデータがエンコードできたことを通知し、ホストCPU101はVUを構成するGOPやAMUの数およびサイズを基にRAM102上のQuickTime管理情報を更新する。

【0210】光ディスク106に記録された直後のQuickTime管理情報における各トラックの状態を図33に示す。ここでは、オリジナルプログラム自体に64kbsでオーディオを記録可能な領域を各PRUから割り当てるとする。このとき、オリジナルトラックにはオリジナルトラックへの割り当てを示すために、既割り当て管理トラックを作成し、そのトラックで各PRU中のオリジナルトラックへ割り当てた領域を管理する。

【0211】一方、移った領域については、未割り当て管理トラックを作成し、そのトラックで管理を行う。さらに、オリジナルトラックに割り当てた領域を管理するため、領域管理トラックを作成する。領域管理トラックに関しては、割り当てられた領域に64kbsでオーディオを記録すると想定し、Sample tableを作成する。

【0212】＜ユーザープログラム作成時の処理＞次に、オリジナルプログラムを外部参照したユーザープログラムを作成する際の手順を説明する。ここでは、作成するユーザープログラムが、図24に示すように、オリジナルプログラムの一群を抜き出して再生というものである場合を考える。

【0213】尚、このユーザープログラムでは、64kbsでアフレコするものとする。また、ユーザによって、ユーザプログラムから参照するオリジナルプログラム中の区

間が決定されていると仮定する。また、オリジナルプログラムに関する管理情報がRAM102上に読み込まれ、新規再生のユーザープログラムに関する管理情報用の領域がRAM102上に確保されていると仮定する。

【0214】まず、RAM102上のオリジナルプログラムに関する管理情報を参照して、未割り当て管理トラックをチェックし、指定された区間が新規に割り当て可能かどうかを調べる。仮に未割り当て領域があった場合、以下

の手順を進める。まず、その領域をこのユーザープログラム用に割り当てる。

【0215】割り当てに伴う未割り当て管理トラックおよび既割り当て管理トラックの変更の様子を図34に示す。割り当てを行う領域を管理している未割り当て管理トラックのチャンクのChunk offsetおよびチャンク(=サンプル)のSample sizeを割り当てに伴い変更し、さらに、既割り当て管理トラックのチャンクのChunk offsetおよびチャンク(=サンプル)のSample sizeを割り当てに伴い変更することで行う。

【0216】次に、指定された区間に対応するビデオトラックおよびメインオーディオトラックのSample tableをユーザープログラム管理情報にコピーする。また、割り当てられた領域に対して領域管理トラックを作成し、64kbsでオーディオを記録すると想定して、Sample tableを作成する。コピーの際にはデータ参照先がオリジナルプログラムになるよう、各トラックのData reference a tomで管理されるデータ参照先をオリジナルプログラムのファイル名に書き換える。

【0217】一方、未割り当て領域がなかった場合、アフレコ用のビットレートを下げることで割り当ててもらう領域のサイズを小さくし、割り当て領域があるかどうかを再度調べる。仮になければ、指定された区間に対応するビデオトラックおよびメインオーディオトラックのSample tableのコピーのみ行う。最後に、RAM102上のオリジナルプログラムの管理情報を光ディスク106に記録する。

【0218】＜アフレコ時の処理＞アフレコ時の処理に関しては、上記第1の実施例と同様であるため、説明を省略する。

【0219】＜静止画付加時の処理＞PRUに記録可能なデータはオーディオデータには限らない。ここでは、既に記録したビデオの特定の区間にスーパーバイーンをするタイトル画像を追加することを想定し、そのタイトル画像をPRUに格納する場合の処理を説明する。

【0220】さらに、上記の例ではアフレコ用の領域は録画時やユーザープログラム再生時に確保していたが、ここではアフレコ時に確保する。尚、以下の処理はオリジナルプログラムに静止面を追加記録する場合を例にとっ

て説明するが、ユーザープログラムであっても処理は共通である。

【0221】すでにタイトル画像がPNG形式で圧縮符号

(16)

29

化され、そのタイトル画像をスーパーバイーンポーズする期間がユーザによって指定されていると仮定する。さらに、静止画追加記録用のオリジナルプログラムの管理情報がRAM102に読み込まれているとする。

【0222】まず、このオリジナルプログラムが静止画用のトラックを持っているかどうかを確認する。持っていないければビデオトラックを1個静止画用に追加する。尚、静止画が動画にスーパーバイーンポーズされるように、track header atom中のlayerの値を動画用のビデオトラックより小さい値に設定する。

【0223】次に、オリジナルプログラムの未割り当て管理トラックを参照してユーザに指定された期間の始端に対応するPRUの領域の空きを調べ、静止面を格納可能かを判断する。格納可能であれば、その静止画データのサイズ分領域の割り当てを行う。尚、割り当ての手順はユーザープログラム作成の場合と同様であるので、説明を省略する。

【0224】また、割り当てた領域に静止面データを記録する。次に、静止面を管理するサンプルを前記静止画用ビデオトラックに追加する。最後に、RAM102上のオリジナルプログラムの管理情報を光ディスク106に記録する。尚、ここでは、領域管理トラックを用いていない。その理由は、前述のオーディオアフレコと異なり、確保しているが未使用という領域がなく、使用管理トラックだけでよいためである。

【0225】尚、ここでは、追加記録時に領域を割り当てたが、ユーザープログラム再生時に割り当ててもよい。ただし、静止面の場合、画像の複雑さによってその符号化データのサイズが異なるため、事前に割り当てるより、エンコードしてサイズが分かってから領域を割り当てた方が利用効率が高い。

【0226】また、ここでは静止面を用いたが、オーディオデータのアフレコの場合にも、ユーザープログラム作成時でなくアフレコ直前に領域を割り当ててもよい。その場合、領域管理トラックは同様に不要になる。

【0227】＜ユーザープログラム削除時の処理＞ユーザーから、ユーザープログラムの削除が指示された場合の処理について説明する。尚、すでに削除指示されたユーザープログラムの管理情報がRAM102上に読み込まれているものとする。

【0228】まず、ユーザープログラムの管理情報中の領域管理トラックを調べ、そのユーザープログラムがアフレコ領域を割り当ててもらっているオリジナルプログラムをリストアップする。

【0229】次に、リストアップした各オリジナルプログラムに随って、以下の処理を行う。まず、オリジナルプログラムの管理情報をRAM102に読み込む。そして、そのオリジナルプログラムから領域を割り当ててもらっているサンプルを含む領域を、オリジナルプログラム中の既割り当て管理トラックから検索する。尚、検索はサン

プル同士のアドレスの比較によって行う。

【0230】検索が終了したら、既割り当て管理トラック中の対象となる領域を開放し、未割り当て領域に返却する。この処理はユーザープログラム割り当ての際の処理の逆に相当する。

【0231】最後に、ユーザープログラムを格納しているファイルや光ディスク106から削除し、オリジナルプログラムの管理情報を光ディスク106に記録する。

【0232】＜バリエーション＞本実施例では、PRU中の複数の未割り当て領域(既割り当て領域)を別トラックに割り当てているが、次のようにすれば未割り当て領域と既割り当て領域をそれぞれ1トラックずつにまとめることも可能である。

【0233】そのためには、PRU中の各領域のdurationの合計を、対応するRUの再生時間と一致させればよい。そうすることで、同一トラック内でRU中の複数の領域を管理することが可能である。

【0234】本実施例と異なり、各領域と対応するRUとの間に時間的なずれが生じることになるが、オリジナルプログラム上での各領域の再生時刻は、対応するRUの再生時刻以上、次のRUの再生時刻未満であり、RUとそれに含まれる領域の間の時間的な対応を致することは可能である。

【0235】また、本実施例では、複数のVUに対応して1個のPRUを置くというAVストリーム構成を用いているが、第1の実施例におけるAVストリーム構成を用いてもよい。さらに、アフレコ用の領域を備えるAVストリームであればどのような形態でも適用可能であることは言うまでもない。

【0236】＜変換例3＞さらに、本発明の第3の実施例について説明する。本実施例は、上述した第1、第2の実施例と類似するため、相違点に就いて以下説明を行う。

【0237】＜AVストリームの形態＞本実施例におけるアフレコ対応AVストリームの構成は、第2の実施例のものと同一であるが、次の点が異なる。後述するようにPRUを整数倍の固定サイズのブロック(PRUブロック)で構成し、PRUブロック単位でプログラムへの領域の割り当てをするため、PRUのサイズがPRUブロックサイズの整数倍となる。

【0238】ここでは、PRUブロックサイズをファイルシステムの論理ブロックサイズと同一の値である2048バイトとする。本実施例におけるAVストリームではRUの先頭にPRUが存在し、RUの先頭はECCブロック境界と一致するため、PRUブロックの境界は論理ブロックの境界と一致する。

【0239】＜AVストリーム管理方法＞AVストリームの管理方法について説明する。VUの管理については第1の実施例と同様であるため、PRUの管理に就いて以下に説明する。

(17)

32

* 的な対応が取れるように、PRUに含まれるサンプリングのdurationの合計が、対応するRUの再生時間と同一になるように、個々のサンプリングのdurationを調整する。

【0243】＜ディスク配置決定方法＞ディスク上でのPRUのサイズをPRブロックの整数倍に合わせるために、次のような変更を行う必要がある。尚、PRUに記録した最大のビットレートを R_0 とし、PRUのサイズはビットレート R_0 のデータを格納可能な最小のサイズにするものとする。

【0244】このとき、第1、第2の実施例と同様、常に＜式1＞を満たせば、リアルタイムアフレコを保証することが可能である。このとき分断ジャンプ時間も含めたRU読み出し時間の最悪値である $T_r(i)$ は、PRブロックの繰上のため、

$$T_r(i) = (T_e(i) \times (R_0 + R_a + R_p) + L_b) / R_s + T_a \dots < \text{式 } 1 >$$

※第3項である。PRUの終端が含まれるECCブロックの記録時間に含まれるためである。

【0246】以上のことから、＜式1＞に＜式1.5＞および＜式1.1＞を代入し、 $T_e(i)$ で解くと、リアルタイムアフレコを保証可能な $T_e(i)$ の条件

$$T_e(i) \geq (3T_a \times R_s + L_b) / (R_s - R_v - R_a - 2R_p) \dots < \text{式 } 1.6 >$$

★上限 T_{emin} は、

$$\dots < \text{式 } 1.7 >$$

【0251】したがって、各RUに対応するPRブロックを管理するチャプターの先頭から18個のData formatをused、それ以外のチャプターのData formatをfreeに設定する。さらに、オリジナルトラックに割り当てた領域を管理するため、領域管理トラックを作成する。領域管理トラックに関しては、割り当てられた領域に64kpbsでオーディオを記録すると想定して、Sample tableを作成する。

【0252】＜ユーザープログラム作成時の処理＞次に、オリジナルプログラムを外部参照したユーザープログラムを作成する際の手順を説明する。ここでは、作成するユーザープログラムが、図24に示すようにオリジナルプログラムの一部を抜き出して再生するというものである場合を考える。

【0253】尚、このユーザープログラムでは64kpbsでアフレコするものと想定する。また、ユーザーによって、ユーザープログラムから参照するオリジナルプログラム中の区間が決定されていると仮定する。また、オリジナルプログラムに属する管理情報がRAM102上に読み込まれ、新規再生のユーザープログラムに属する管理情報の領域がRAM102上に確保されていると仮定する。

【0254】まず、RAM102上のオリジナルプログラムに属する管理情報を参照し、割り当て管理トラックをチェックして、指定された区間に対し、領域を新規に割り当

37

て可能かどうかを調べる。具体的には、オリジナルプログラム中の各RUから割り当てらる必要のあるPRブロック数を計算し、それだけの数のfreeのPRブロックが各RUに連続して存在するかどうか調べる。

【0255】仮に存在した場合、以下の手順を進める。まず、その領域をこのユーザープログラム用に割り当て、その領域をこのユーザープログラム用に割り当てる。割り当ては、割り当ててもらった領域を管理するチャプターのData formatをusedに変更することで行う。次に、指定された区間に対応するビデオトラックおよびオーディオトラックのSample tableをユーザープログラム管理情報にコピーする。

【0256】また、割り当てられた領域に対して領域管理トラックを作成し、64kpbsでオーディオを記録すると想定して、Sample tableを作成する。コピーの際にはデータのData reference atomで管理されるデータ参照先をオリジナルプログラムのファイル名に書き換える。

【0257】一方、未割り当て領域がなかった場合、アフレコ用のビットレートを下げることで割り当ててもらった領域のサイズを小さくし、割り当て領域があるかどうかを再度調べる。仮になければ、指定された区間に対応するビデオトラックおよびメインオーディオトラックのSample tableのコピーのみ行う。最後に、RAM102上のオリジナルプログラムおよびユーザープログラムの管理情報を光ディスク106に記録する。

【0258】＜アフレコ時の処理＞アフレコ時の処理に関しては、第1の実施例と同様であるため、説明を省略する。

【0259】＜静止画面付加時の処理＞本実施例でも、第2の実施例と同様、静止画面等のデータを付加することや、領域の割り当てをユーザープログラム作成時でなく、データ記録時に行うことは可能である。第2の実施例との違いは、割り当て管理トラックの構成によって生じる違いであり、その違いについては、＜ユーザープログラム作成時の処理＞に記載してあるため、ここでは説明を省略する。

【0260】＜ユーザープログラム削除時の処理＞ユーザーから、ユーザープログラムの削除が指示された場合の処理について説明する。尚、すでに、削除指示されたユーザープログラムの管理情報がRAM102上に読み込まれているものとする。

【0261】まず、ユーザープログラムの管理情報中の領域管理トラックを調べ、そのユーザープログラムがアフレコ領域を割り当ててもらっているオリジナルプログラムをリストアップする。

【0262】次に、リストアップした各オリジナルプログラムに関して、以下の処理を行う。まず、オリジナルプログラムの管理情報をRAM102に読み込む。次に、そのオリジナルプログラムから領域を割り当ててもらっているサンプルを含む領域を、オリジナルプログラム中の割

(18)

34

り当て管理トラックから検索する。

【0263】尚、検索はサンプリング時のアドレスの比較によって行う。検索が終了したら、割り当て管理トラック中の対象となるチャプターのData formatをfreeに変える。最後に、ユーザープログラムを格納しているファイルに光ディスク106から削除し、オリジナルプログラムの管理情報を光ディスク106に記録する。

【0264】＜バリエーション＞本実施例では、従来のVUに対応して1個のPRUを置くというAVストリーム構成を用いているが、第1の実施例におけるAVストリーム構成を用いてもよい。さらに、アフレコ用の領域を備えるAVストリームであればどのような形態でも適用可能であることは言うまでもない。

【0265】また、本実施例では、未割り当て領域の割り当ての際にData formatを利用したが、区別が可能であれば別種の情報を使っても構わないことは言うまでもない。例えば、各サンプル(=PRブロック)が未割り当てか既割り当てかを示す独自の管理情報を、割り当て管理トラックのSample table atomに記入することも考えられる。

【0266】また、未割り当て領域を区別するのには割り当て管理トラックのEdit list atomを用いる方法も、本実施例のバリエーションとして考えられる。具体的には、Edit list atomを用いて割り当て済みのサンプル(=PRブロック)を再生対象に設定し、それ以外のサンプルを再生対象から外すことで、未割り当てか既割り当てかを区別するというものである。

【0267】さらに、本実施例では、録画時やユーザープログラム作成時に各プログラムに領域を割り当てているが、前述の第2の実施例で説明したように、アフレコ時に割り当ててもよい。その場合、各プログラムに割り当てられている領域は既にデータが記録されることになるため、領域管理トラックを備える必要はない。

【0268】＜実施例4＞次に、本発明の第4の実施例について説明する。本実施例は、上述の第1乃至第3の実施例と類似するため、相違点に絞って以下説明を行う。

【0269】＜AVストリームの形態＞本実施例におけるAVストリームの形態は、第2の実施例のものと同じであるため、説明を省略する。

【0270】＜AVストリーム管理方法＞AVストリームの管理方法について説明する。VUの管理については、第1の実施例と同様であるため、PRUの管理に絞って以下に説明する。

【0271】第1の実施例と同様、本実施例でもPRUを管理するために割り当て管理トラックと領域管理トラックとを使用管理トラックを用いる。領域管理トラックと使用管理トラックについては、第1の実施例と同様であるため、割り当て管理トラックについての説明する。

【0272】割り当て管理トラックは、図36に示すよ

37

【0240】第1の実施例と同様、本実施例でもPRUを管理するために、割り当て管理トラックと領域管理トラックとを使用管理トラックとを用いる。領域管理トラックと使用管理トラックについては、第1の実施例と同様であるため、割り当て管理トラックについての説明する。

【0241】割り当て管理トラックは、第1の実施例や第2の実施例と異なり、1トラックで構成される。前述のPRブロック単位に、ユーザープログラムへの領域の割り当てを管理する。具体的には、図35に示すように、前述のPRブロック1個を1サンプルおよび1チャプターに対応させ、チャプターのData formatで、領域が未割り当てか既割り当てかを区別できるようにする。

【0242】そのために、Data formatに独自の種別をfree、usedを定義する。また、PRUとRUとの間の時間* $T_r(i) = (T_e(i) \times (R_0 + R_a + R_p) + L_b) / R_s + T_a$ となる。

【0245】ここで、Lは1個のPRブロックのサイズを表す。一方、 $T_e(i)$ は第2の実施例における＜式1.1＞をそのまま使うことが可能である。なぜなら、ここではPRブロックのサイズを論理ブロックサイズと等しくしているため、PRブロックの繰上げ分は、＜式1.1＞の右辺※

$$T_e(i) \geq (3T_a \times R_s + L_b) / (R_s - R_v - R_a - 2R_p)$$

が得られる。

【0247】つまり、アフレコ保証可能な再生時間下★ $T_{\text{emin}} = (3T_a \times R_s + L_b) / (R_s - R_v - R_a - 2R_p)$ となる。

【0248】このとき、再生時間の上限 T_{max} を次のように設定する。上限値を設定するのは、第1の実施例と同様の理由に基づき、

$$T_{\text{max}} = T_{\text{emin}} + T_{\text{max}} \dots < \text{式 } 1.8 >$$

＜記録時の処理＞記録するAVストリームは、ビデオのビットレート $R_v = 5\text{kpbs}$ 、メインオーディオのビットレート $R_p = 25\text{kpbs}$ で、PRUに記録する最大のビットレート $R_0 = 25\text{kpbs}$ のアフレコ対応ストリームとしたとき、＜式1.7＞に＜式1.8＞に $R_s = 20\text{kpbs}$ 、 $T_a = 1\text{秒}$ 、 $T_{\text{max}} = T_{\text{emin}} = 約0.5\text{秒}$ を代入し $T_e(i)$ の範囲4.23秒以上4.73秒以下が得られる。

【0249】 T_{max} は約0.5秒でこの条件を満たすのは、 $T_e(i) = 4.5\text{秒}$ のとき異なり、9個のVU毎でRUを構成することになる。このときPRUのサイズは、 $\text{ceiling}(4.5\text{秒}) \times 256\text{kpbs} / 2\text{KB}$ となり、71個のPRブロックで構成されることになる。記録時の処理は、第2の実施例と同様であるため、説明を省略する。

【0250】光ディスク106に記録された直後のQuickTime管理情報における各トラックの状態について説明する。ここでは、オリジナルプログラム自体に64kpbsでオーディオを記録可能な領域を各RUから割り当てたとする。このとき、割り当て管理トラックのデータは(4.5秒)×64(kpbs)/2KB、すなわち、18ブロックとなる。

(19)

35

うに、1トラックで構成され、そのトラックはPRU中の既割り当て領域および未割り当て領域を問わず、すべての領域をチャッキングして管理する。また、各領域とRUの時間的対応が取れるよう、各PRU中のチャッキングのdurationの合計が、各PRUを含むRUのdurationと一致するようにする。

【0273】また、各チャッキングの管理している領域が既割り当て領域か未割り当て領域かを区別できるよう、第3の実施例と同様、チャッキングのData formatに種別'free'、'used'を独自に定義し、割り当て状況に応じてData formatにいずれかを指定する。

【0274】<ディスク配置決定方法>ディスク上でのチャッキングの状態について説明する。オリジナルプログラムのには、ビデオトラック、メインオーディオトラックの他に、割り当て管理トラックおよび領域管理トラックが存在する。ここでは、オリジナルプログラム自体に64kbitsでオーディオを記録可能な領域を各RUから割り当てると仮定する。

【0277】第2の実施例と同様、AVストリームを構成するRUの再生時間が4.5秒だとすると、 $4.5 \times 64kbits = 36000$ バイトの領域をオリジナルプログラムに割り当てることとなる。つまり、各PRUの先頭から36000バイトの領域がそれぞれData formatが'used'のチャッキングで管理され、PRU残りの領域はそれぞれData formatが'used'のチャッキングで管理される。

【0278】さらに、オリジナルトラックに割り当てた領域を管理するため、領域管理トラックを作成する。領域管理トラックに関しては、割り当てられた領域に64kbitsでオーディオを記録すると想定して、Sample tableを作成する。

【0279】<ユーザプログラム作成時の処理>次に、オリジナルプログラムを外部参照したユーザプログラムを作成する際の手順を説明する。ここでは、作成するユーザプログラムが、図24に示すように、オリジナルプログラムの一部を抜き出して再生するというものであることを考える。

【0280】尚、このユーザプログラムでは64kbitsでアフレコするものとする。また、ユーザによって、ユーザプログラムから参照するオリジナルプログラム中の区間が既に指定されているものと仮定する。さらに、オリジナルプログラムに関する管理情報がRAM102上に読み込まれ、新規作成するユーザプログラムに関する管理情報を用いて、新設されていると仮定する。

【0281】まず、RAM102上のオリジナルプログラムに関する管理情報を参照し、割り当て管理トラックを参照して、オリジナルプログラムにおける領域の管理情報を示す説明図である。

【0282】また、本実施例では、録画時やユーザプログラム作成時に各プログラムに領域を割り当てていくが、前述の第2の実施例で説明したように、アフレコ時に割り当ててもよい。その場合、各プログラムに割り当てられている領域は既にデータが記録されることになるため、領域管理トラックを備える必要はない。

【0297】

【発明の効果】以上説明したように、本発明によれば、アフレコ用領域割り当てを管理する情報を備えることで、複数のユーザプログラムからアフレコした場合、意図せず書きをすることを防ぐことが可能になる。

【図面の簡単な説明】

【図1】本発明の実施形態における領域構成を示すブロック図である。

【図2】QuickTimeファイルフォーマットにおける管理情報とAVストリームの関係を示す説明図である。

【図3】QuickTimeファイルフォーマットにおけるMovie atomの概要を示す説明図である。

【図4】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

【図5】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

【図6】QuickTimeファイルフォーマットにおけるMedia atomの構成を示す説明図である。

【図7】QuickTimeファイルフォーマットにおけるMedia information atomの構成を示す説明図である。

【図8】Sample table atomによるデータ管理の例を示す説明図である。

(20)

37

RAM102に読み込む。次に、そのオリジナルプログラムから領域を割り当ててもらっているサンプルを含む領域を、オリジナルプログラム中の割り当て管理トラックの'used'のチャッキングから検索する。尚、検索はサンプルのアドレスの比較によって行う。検索が終了したら、対象となる'used'チャッキングの領域を縮小し、その分、'free'チャッキングの領域を拡大する。

【0292】最後に、ユーザプログラムを格納しているファイルに光ディスク106から削除し、オリジナルプログラムの管理情報を光ディスク106に記録する。

【0293】<バリエーション>本実施例では、複数のRUに対して個々のPRUを置くというAVストリーム構成を用いているが、第1の実施例におけるAVストリーム構成を用いてもよい。さらに、アフレコ用の領域を備えるAVストリームであればどのような形態でも運用可能であることは言うまでもない。

【0294】また、本実施例では、未割り当てか既割り当てか別のData formatを利用したが、区別が可能であれば別種の情報を使っても構わないのは言うまでもない。例えば、各チャッキングが未割り当てか既割り当てを示す独自の管理情報を、割り当て管理トラックのSample table atomに導入することも考えられる。

【0295】さらに、未割り当てか既割り当てかを区別するに割り当て管理トラックのEdit list atomを用いる方法も、本実施例のバリエーションとして考えられる。具体的には、Edit list atomを用いて割り当て済みのチャッキングを再生対象に設定し、それ以外のチャッキングを再生対象から外すことで、未割り当てか既割り当てかを区別するというものである。

【0296】そしてまた、本実施例では、録画時やユーザプログラム作成時に各プログラムに領域を割り当てていくが、前述の第2の実施例で説明したように、アフレコ時に割り当ててもよい。その場合、各プログラムに割り当てられている領域は既にデータが記録されることになるため、領域管理トラックを備える必要はない。

【0297】

【発明の効果】以上説明したように、本発明によれば、アフレコ用領域割り当てを管理する情報を備えることで、複数のユーザプログラムからアフレコした場合、意図せず書きをすることを防ぐことが可能になる。

【図面の簡単な説明】

【図1】本発明の実施形態における領域構成を示すブロック図である。

【図2】QuickTimeファイルフォーマットにおける管理情報とAVストリームの関係を示す説明図である。

【図3】QuickTimeファイルフォーマットにおけるMovie atomの概要を示す説明図である。

【図4】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

【図5】QuickTimeファイルフォーマットにおけるTrack atomの概要を示す説明図である。

【図6】QuickTimeファイルフォーマットにおけるMedia atomの構成を示す説明図である。

【図7】QuickTimeファイルフォーマットにおけるMedia information atomの構成を示す説明図である。

【図8】Sample table atomによるデータ管理の例を示す説明図である。

【図9】QuickTimeファイルフォーマットにおけるSample table atomの構成を示す説明図である。

【図10】QuickTimeファイルフォーマットにおけるEdit list atomの構成を示す説明図である。

【図11】Edit list atomによる再生範囲指定の例を示す説明図である。

(20)

38

header atomの構成を示す説明図である。

【図6】QuickTimeファイルフォーマットにおけるMedia atomの構成を示す説明図である。

【図7】QuickTimeファイルフォーマットにおけるMedia information atomの構成を示す説明図である。

【図8】Sample table atomによるデータ管理の例を示す説明図である。

【図9】QuickTimeファイルフォーマットにおけるSample table atomの構成を示す説明図である。

【図10】QuickTimeファイルフォーマットにおけるEdit list atomの構成を示す説明図である。

【図11】Edit list atomによる再生範囲指定の例を示す説明図である。

【図12】QuickTimeファイルフォーマットにおけるUse r-defined data atomの構成を示す説明図である。

【図13】本発明の第1の実施例におけるストリームの構成を示す説明図である。

【図14】本発明の第1の実施例におけるのアフレコ非対応VUの構成を示す説明図である。

【図15】本発明の第1の実施例におけるアフレコ対応VUの構成を示す説明図である。

【図16】本発明の第1の実施例におけるQuickTimeによるAVストリーム管理形態を示す説明図である。

【図17】本発明の第1の実施例におけるオリジナルプログラムの管理形態を示す説明図である。

【図18】本発明の第1の実施例におけるユーザプログラムの管理形態を示す説明図である。

【図19】本発明の第1の実施例におけるTrack propertyの構成を示す説明図である。

【図20】本発明の第1の実施例におけるリファレンス・デハイス・モデルを示す説明図である。

【図21】本発明の第1の実施例におけるリファレンス・アフレコ・アルゴリズムを示す図である。

【図22】本発明の第1の実施例における記録動作を示すフローチャートである。

【図23】本発明の第1の実施例における記録直後のオリジナルプログラムの管理形態を示す説明図である。

【図24】本発明の第1の実施例における作成直後のユーザプログラムの管理形態を示す説明図である。

【図25】本発明の第1の実施例におけるユーザプログラム作成直後のオリジナルAVストリームの管理形態を示す説明図である。

【図26】本発明の第1の実施例におけるアフレコ動作を示すフローチャートである。

【図27】本発明の第1の実施例におけるアフレコ直後のオリジナルプログラムの管理形態を示す説明図である。

【図28】本発明の第1の実施例におけるアフレコ直後のユーザプログラムの管理形態を示す説明図である。

【図29】本発明の第1の実施例における再生動作を示す説明図である。

【図30】本発明の第1の実施例における再生動作を示す説明図である。

【図31】本発明の第1の実施例における再生動作を示す説明図である。

(21)

39

ソフトウェアである。

【図30】本発明の第2の実施例におけるストリームの構成を示す説明図である。

【図31】本発明の第2の実施例における割り当て管理トラックの概要を示す説明図である。

【図32】本発明の第2の実施例におけるリファレンス・アフレコ・アルゴリズムを示す説明図である。

【図33】本発明の第2の実施例における記録直後のオリジナルプログラムの管理情報を示す説明図である。

【図34】本発明の第2の実施例におけるユーザプログラム作成時の割り当て管理トラックの変更を示す説明図である。

【図35】本発明の第3の実施例における割り当て管理トラックの概要を示す説明図である。

【図36】本発明の第4の実施例における割り当て管理トラックの概要を示す説明図である。

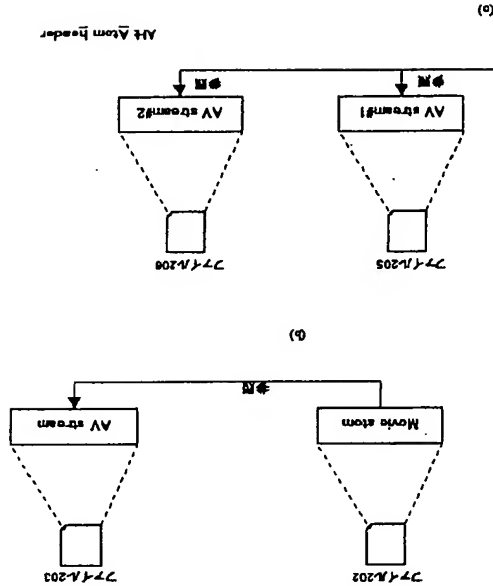
【図37】本発明の第4の実施例におけるユーザプログラム作成時の割り当て管理トラックの変更を示す説明図である。

【図38】従来技術におけるディスク上での記録形態を示す説明図である。

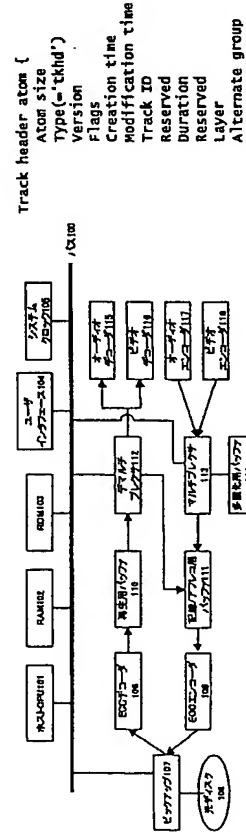
【図39】従来技術におけるアフレコ時のヘッドの動き

(22)

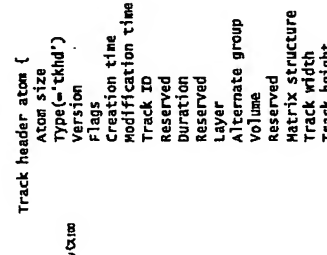
【図2】



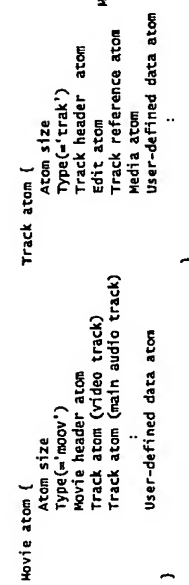
【図1】



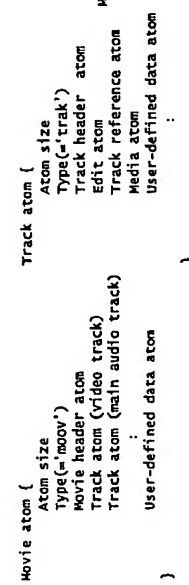
【図5】



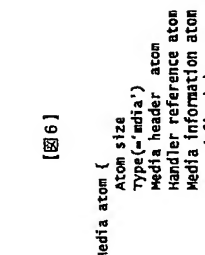
【図3】



【図4】

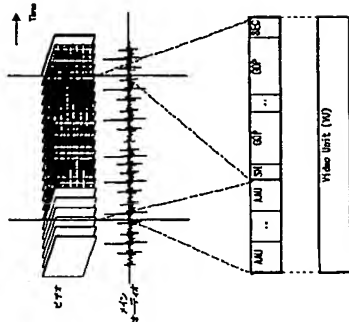


【図6】

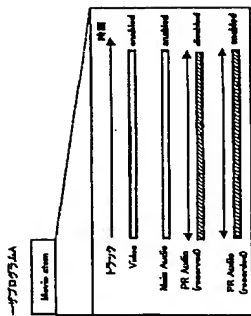


(25)

【図14】

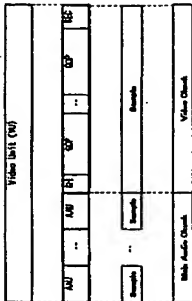


【図18】

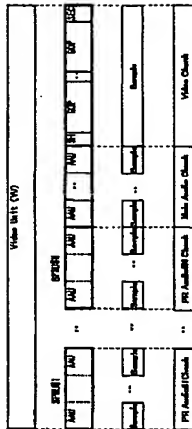


(26)

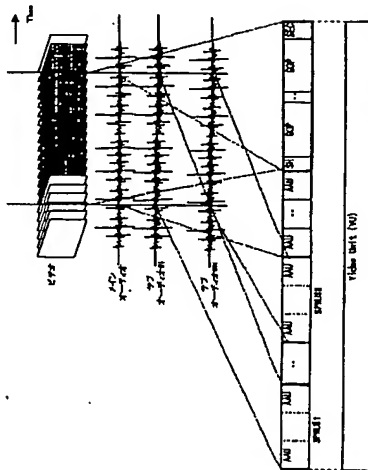
【図16】



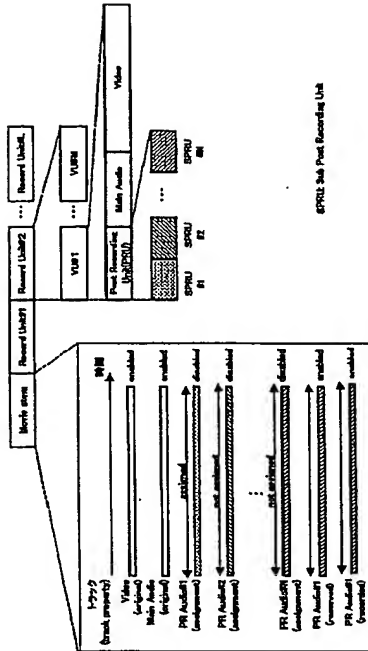
(27)



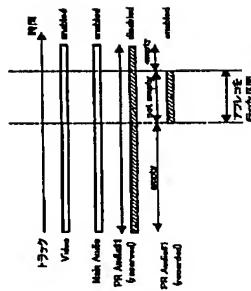
【図15】



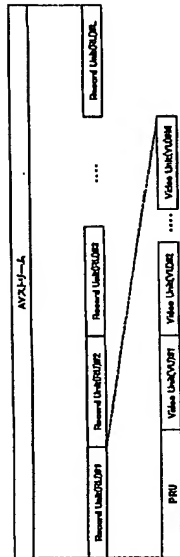
【図17】



【図28】



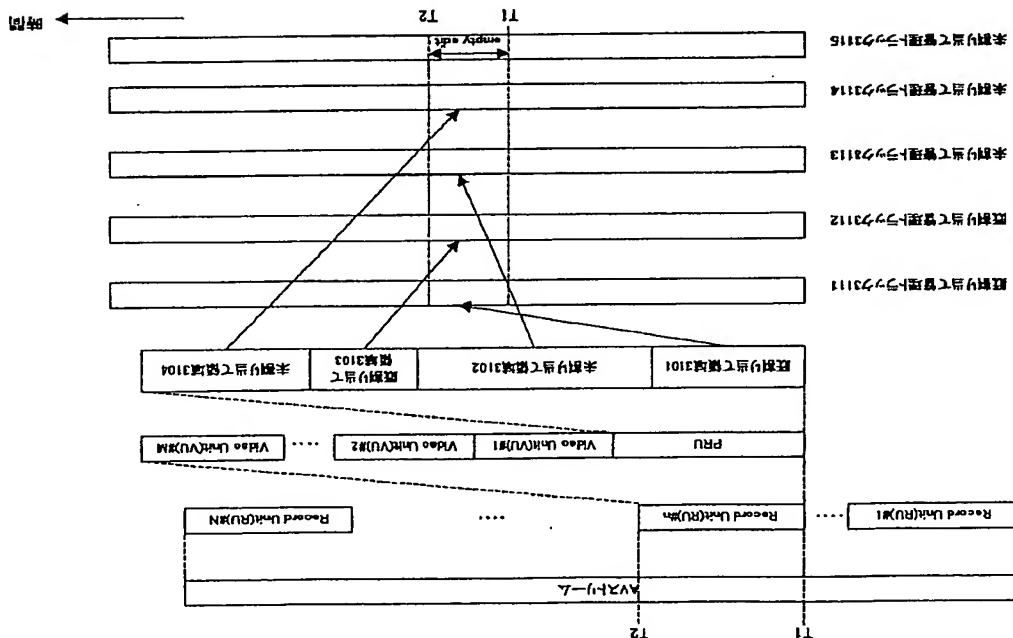
【図30】



SI: Sequence Header
ES: Sequence End Code
SPS: Sub Picture Descriptive Unit

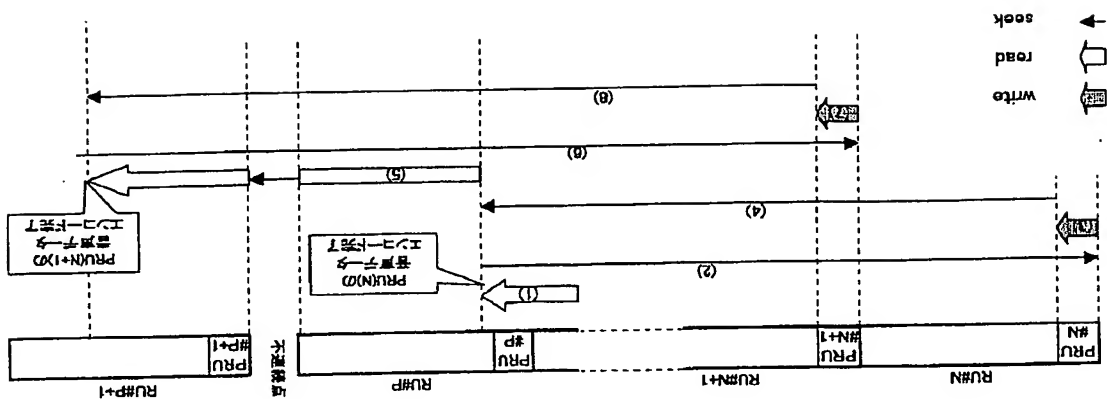
(31)

【図 3.1】



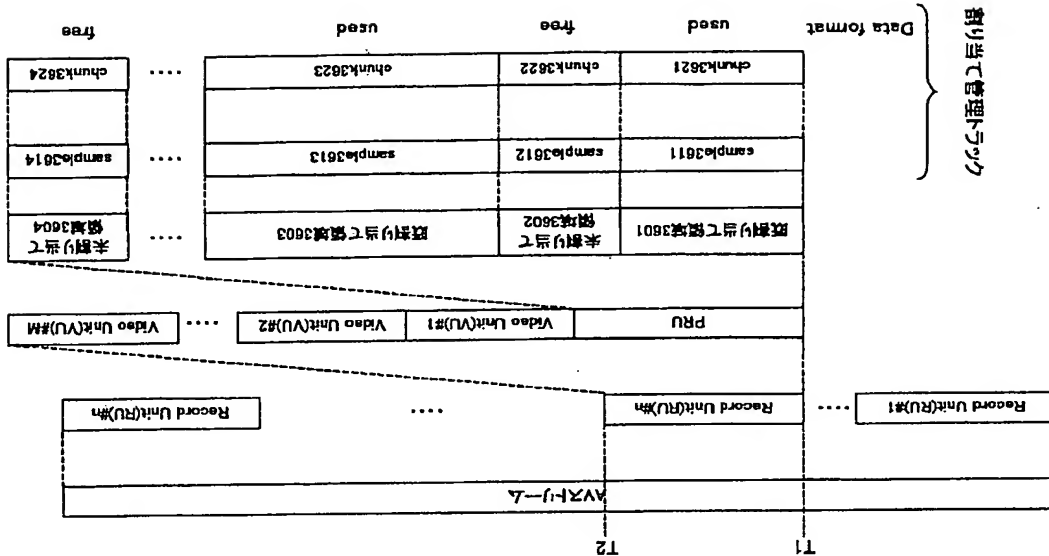
(32)

【図 3.2】



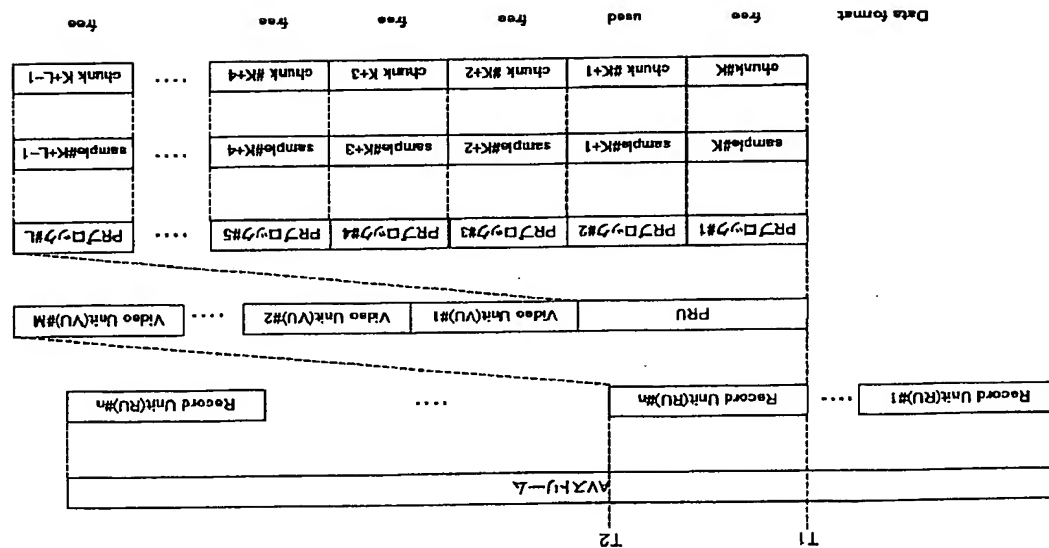
(34)

【図 3 6】



(33)

【図 3 5】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)